

1) Implementa una funcionalidad para recibir distintos tipos de vehículos y realizar inspecciones personalizadas según el tipo de vehículo.

Describe cómo los objetos se relacionan para completar una revisión exhaustiva en el taller.

Incluye un diagrama de clases que muestre la relación entre el taller, los vehículos, las partes revisadas y los mecánicos.

Define al menos 3 métodos clave que los objetos se envían entre sí para realizar la inspección de cada tipo de vehículo, considerando las diferencias de inspección según el tipo de vehículo (por ejemplo, `revisarMotor()` en Automóvil o `revisarCadena()` en Motocicleta).

Explica los métodos principales de cada clase para que el sistema de revisión funcione (por ejemplo, `asignarMecánico()` en Taller, `realizarRevisión()` en Mecánico).

Implementa la funcionalidad básica para que el taller pueda recibir un vehículo, inspeccionar sus partes, y reportar el estado de la revisión.

3)Crea una funcionalidad para recibir distintos tipos de visitantes y verificar si cumplen con los requisitos específicos de cada atracción, como altura, edad y peso. Explica cómo los objetos se relacionan para gestionar la entrada a cada atracción.

Incluye un diagrama de clases que muestre la relación entre el parque, las atracciones, los requisitos específicos y los operadores de atracciones.

Define al menos 3 métodos clave que los objetos se envían entre sí para realizar las verificaciones de acceso, considerando las variaciones según el tipo de atracción (por ejemplo, `verificarAltura()` en MontañaRusa o `verificarEdad()` en CasaDelTerror).

Describe los métodos principales para que el sistema de acceso funcione correctamente (por ejemplo, `asignarOperador()` en Parque, `verificarAcceso()` en Operador).

Implementa la funcionalidad para que el parque pueda recibir visitantes, realizar verificaciones de acceso y reportar si el visitante puede ingresar a la atracción.

